

27
150 pgs.

AVON

Enciclopedia Práctica del Spectrum



Nueva Lente/Ingelek



PRINCIPIOS GENERALES DEL SONIDO



El sonido se produce cuando un elemento físico perturba la atmósfera, creando variaciones en la presión del aire que le rodea. Dichas variaciones de presión se propagan en forma de onda esférica y, si son interceptadas por la membrana del tímpano, se transforman en impulsos nerviosos que van al cerebro, donde se interpreta la información recibida.

Si la presión generada pasa momentáneamente a su valor máximo, y permanece en él durante un breve intervalo de tiempo, volviendo bruscamente al valor mínimo, entonces se califica a la onda como «cuadrada», a diferencia de la ya conocida y más suave onda «senoidal».

Las ondas armónicas son tanto aquellas producidas por medios naturales, como las emitidas por los instrumentos musicales. A este tipo de ondas podemos denominarlas también, atendiendo a la forma en que son producidas, «análogas».

Decimos que una onda es de tipo analógico o sinusoidal, cuando el paso del valor máximo al mínimo se produce de una forma gradual, siendo su gráfica una curva continua, similar a la de la función seno.

Por el contrario, las ondas de tipo cuadrado o «digital» son los producidos por osciladores electrónicos. Se trata pues de un sonido artificial, en el cual el paso del valor máximo al mínimo se efectúa de una forma brusca, produciendo una onda cuya imagen es similar a las almenas de un castillo.

CARACTERÍSTICAS DE LAS ONDAS

Hagamos un viaje a través de las ondas para conocer sus características, en todo momento, nos apoyaremos en los gráficos adjuntos para llegar a comprender este nuevo vocabulario.

Si consideramos las gráficas de los dos tipos de ondas, las ordenadas (eje vertical) expresarán el «pulso» y las abscisas (eje horizontal) el «tiempo», denominándose «crestas» los picos de mayor altura y «valles» los de menor.



El sonido se produce cuando un elemento físico perturba la atmósfera.

De esta forma, se denomina «amplitud» de onda a la máxima distancia alcanzada entre una cresta o un valle y el eje de abscisas, y «longitud» de onda a la distancia existente entre dos crestas o dos valles consecutivos.

El sonido se propaga en forma de ondas esféricas, que serán ser interceptadas por la membrana del tímpano.





La onda cuadrada es un sonido artificial. Se produce cuando el punte del teclado empuja al mecanismo de de ferrocromo.

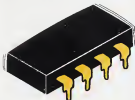
El tono de un sonido puede ser más grave o más agudo. Con mayor propiedad, podemos decir que el tono está en relación directa con la frecuencia del sonido producido, pudiéndose definir ésta como la cantidad de variaciones completas que tienen lugar en un segundo, es decir, el número de longitudes de onda descritas en la unidad de tiempo, medida en Hertzios (Hz) o ciclos por segundo (C/S).

La gama de tonos audibles por el ser humano va desde los 40 a los 15.000 Hz., aproximadamente, mientras que el Spectrum puede producir una serie de tonos de onda cuadrada, situados entre 10 y 15.000 Hz., aproximadamente. La forma de

generar dichos sonidos consiste en conmutar y desconmutar electrónicamente, a intervalos regulares, una señal de tensión constante, y llevar las variaciones amplificadas a un pequeño altavoz alojado en su interior.

El control del sonido, tanto en la duración como en la frecuencia, se efectúa en el BASIC del Spectrum por medio de la sentencia **BEEP D, N**, donde el parámetro **D** representa la duración en segundos, y **N** el código del tono o frecuencia. Este último puede tomar cualquier valor entero comprendido entre -59 (unos 10 Hz.) y +69 (15.000 Hz. aproximadamente).

Concretamente, cuando **N** sea igual a cero, la frecuencia generada corresponde al tono de 0a octava del de la escala del piano, sobre 261 ciclos. La escala completa se consigue, internamente, utilizando las siguientes frecuencias.



BEEP D, N

El control del sonido lo efectúa el C. P. U. del Spectrum por medio de BEEP D, N.

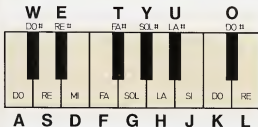
Los sonidos musicales son producidos por medio de instrumentos musicales.



INT. NOTA FRECUENCIA

C	DO	261.63
C#	DO#	277.18
D	RE	293.66
D#	RE#	311.13
E	Mi	329.63
F	FA	349.23
F#	FA#	369.99
G	SOL	392.00
G#	SOL#	415.30
A	LA	440.00
A#	LA#	466.16
B	SI	493.88

Donde la columna "INT." corresponde a la notación internacional y el símbolo "#" al "sostenido" de la nota correspondiente, como explicaremos más adelante.



Correspondencia entre las notas musicales y las letras utilizadas por el ordenador, y su posición en el teclado

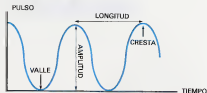
La gama de tonos audibles por el ser humano va desde 0Hz a 15.000 Hz. El Spectrum produce una gama de tonos de onda cuadrada entre 10 y 15.000 Hz.



La diferencia entre el lenguaje musical y el del Spectrum es únicamente cuestión de formas, como ya veremos. Pero, antes de nada, conozcamos los componentes elementales de la música, las notas, cuya representación escrita nos indica la altura del tono en que nos encontramos y la duración del mismo.

Para una más fácil comprensión de los no incómodos, consideraremos el teclado de un piano. En

Representación gráfica de la onda sinusoidal y cuadrada con sus principales características.



COMPONIENDO MUSICA

La música se compone fundamentalmente de sonido y ritmo, es decir, de la producción de ciertos tonos durante tiempos determinados.





El símbolo # representa el sostenido de la nota correspondiente.



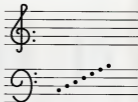
El teclado un conjunto de teclas blancas centrales, correspondientes a las notas llamadas Do, Re, Mi, Fa, Sol, La y Si (C, D, E, F, G, A, B en nomenclatura internacional), que constituyen una «octava», y a las que corresponden ciertos tonos o frecuencias y unos códigos de tono N determinados, como ya hemos visto anteriormente.

En dicho teclado existen además otras teclas negras, correspondientes a los tonos intermedios que quedan por definir. Estos tonos intermedios se pueden apellidar «sostenido» (♯) y adoptar el nombre de la nota anterior, o bien «bemol» (♭) si el nombre adoptado es el de la nota posterior. Por ejemplo, el tono de Do ♯ (Do sostenido) coincide con el de Re ♭ (Re bemol). Cabe añadir que la distancia entre dos tonos consecutivos (dos teclas contiguas blancas o blanca y negra), se denomina «semitono».

La escritura de las notas se realiza sobre un conjunto de cinco líneas, llamado pentagrama. En nuestro caso utilizamos dos de sus muchas versiones: el de clave de Sol en segunda línea y el de clave de Fa en cuarta línea.

Para hacer más una idea más clara de todo lo dicho, veremos a continuación un programa generador de la escala musical.

REM - ESCALA MUSICAL J. M. LOPEZ MARTINEZ
FOR I = 1 TO 8: READ N: BEEP, 25, N: NEXT N
DATA 0, 2, 4, 6, 7, 8, 1, 12



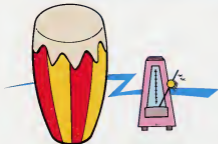
La escritura de las notas se realiza en un pentagrama. Las claves más utilizadas son la de Sol y la de Fa.

DURACION DE LAS NOTAS

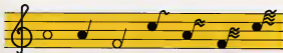
Si no existe el debido control de la duración de cada nota no puede haber ritmo, ni por lo tanto música. Esta duración se indica junto a la propia nota, mediante un dibujo que se añade a la misma. La figura generalmente aceptada como referencia es la «negra», y su duración es aproximadamente de 1/4 de segundo. Las restantes figuras de duración se denominan «redonda», «blanca», «corchea», «semicorchea», «fusa», etc.

De esta manera, cada sonido queda completamente determinado por su duración, mediante un dibujo adicional, y por su altura de tono, mediante su posición en el pentagrama.

Por lo tanto, la transcripción a los parámetros D y N resulta trivial y puramente mecánica. No obstante, para facilitar aún más la labor, podemos tomar como referencia el valor de la duración de la fusa, que es 1/32 de segundo. De esta forma, los valores de las duraciones del resto de las notas serán las sucesivas potencias de dos, lo cual facilitará en gran medida la labor de programación. El puntillo colocado a la derecha de cualquier nota aumenta su duración en la mitad, y los as-



La música se compone fundamentalmente de sonido y ritmo.



REDONDA NEGRA BLANCA CORCHEA SEMICORCHEA FUSA SEMIFUSA

Mediante estas figuras puede determinarse la duración del sonido, y la altura del tono por su posición en el pentagrama.

lancas son los tiempos muertos en los cuales el compositor ha decidido que no debe haber sonido alguno. Existen dos alternativas para interpretar los silencios y, como es obvio, ambas utilizan la sentencia PAUSE.

La primera consiste en poner un signo - en la duración, que será directado con una comparación. La segunda es añadir un tercer conjunto de dígitos de entrada, de manera que si el dato leído es cero, la comparación lo ignore y genere el tono, de no ser así (si no es cero), el ordenador estará

en presencia de un silencio y desvarará el control hacia la sentencia PAUSE correspondiente.

El único y mínimo inconveniente es que PAUSE funcione en unidades de 1/50 de segundo, y la coincidencia de duraciones con los tonos puede no ser absoluta.

EMPEZANDO A COMPOSER

Con los conocimientos básicos que hasta ahora hemos adquirido, estamos en condiciones de construir un programa capaz de emitir sonidos como un piano, que nos permita componer melodías.

Nuestro piano, consta de 8 teclas blancas (Do, Re, Mi, Fa, Sol, La, Si, Do), localizadas en el teclado del Spectrum sobre la línea media de las letras (A, S, D, F, G, H, J, K, L) y de 6 teclas negras (Do, Re, Fa, Sol, La, Do), situadas en W, E, T, Y, U y O de la fila superior de letras. De cualquier forma, nuestro piano es generosamente didáctico, y nos va a mostrar en pantalla la distribución de las teclas. Además, cada vez que pulsamos una, nos lo indicará a través de señales visuales para orientarnos sobre la posición de los dedos.

Para interpretar las acciones anteriores la sentencia PAUSE.



```

38 SON = PIANO - J.M. LOPEZ MARTINEZ
39 END PROC
40 FOR I=1 TO 20 READ M(I), N(I)
41 FOR J=L TO R
42 PRINT AT J+1,30+I
43 FOR J=L TO P
44 IF J=O OR J=U THEN PRINT " "
45 PRINT " "
46 NEXT J
47 NEXT I
48 PRINT AT 6,10 TO R F S L
49 AT 7,10 TO R S T P U
50 I OVERWRITE LA,VA, S, D, F, G, H, J
51 L, W, E, T, Y, U W N F E L S O
52
53 FOR I=1 TO 81 PRINT LA+I*20,00-20*I
54 AT 20,00: DRAW P,VA: DRAW -20,00: DRAW 0, 40
55 NEXT I
56 LET X0=100000: IF X0<< THEN GO TO 100
57 IF X0<100000 AND X0<100000 AND X0<100000
58 AND X0<100000 AND X0<100000 AND X0<100000
59 AND X0<100000 AND X0<100000 AND X0<100000
60 AND X0<100000 AND X0<100000 AND X0<100000

```

!

Se dice que una onda es cuadrada cuando la presión para mantenerse en su valor máximo, y permanece con él durante un breve intervalo de tiempo, volviendo bruscamente al valor mínimo.

*

Las ondas de tipo cuadrado o «digital» son las producidas por los circuitos electrónicos. En estos, el paso del voltaje mínimo al máximo se efectúa de una forma brusca, produciendo una onda cuya amplitud sea similar a la de las alteraciones de un cuadro.

*

El sonido se produce cuando un elemento físico comienza la vibración, creando un movimiento en la presión del aire que lo rodea.



!

En nuestro piano, cada vez que pulsamos una tecla, se le indica, para orientarse sobre la posición de los dedos

imprima por medio de la función DRAW, los contornos de las teclas «blancas»

En el bloque formado por las instrucciones 130 y 140, se aceptan caracteres del teclado por medio de la sentencia INKEY\$, eliminándose en un primer paso las respuestas nulas y en un segundo las pulsaciones de todas aquellas teclas diferentes de las reseñadas como utilizables en el teclado del piano



El sonido se produce a través de la sentencia BEEP.

En la línea 150 se continúa con el tratamiento del carácter pulsado, ya forzosamente correcto, produciéndose el sonido propiamente dicho, a través de las sentencias BEEP, con una duración fija de un cuarto de segundo de la nota correspondiente en la matriz N(). Al comienzo y final de esta línea, antes de volver a la posición de la

Las ondas asociadas a las son las producidas por medios naturales, y se denominan «vibración armónica». En ellas, el peso del volante máximo al resaca se produce de una forma gradual, siendo su gráfico una curva sinuosa, similar a la gráfica de la función seno

*

La distancia entre dos tonos consecutivos (dos teclas contiguas blancas o negras y no gris) se denomina «semitono»

*

Existen unas teclas negras que corresponden a los tonos intermedios. Cada tanto de la nota anterior, o «bemol» (b) si el nombre adoptado es el de la nota posterior

```
AND X%(1) AND X%(1) THEN GO TO 136
135 GO SUB 100: BEEP: 25,80000: X%-1
GO SUB 200: GO TO 138
136 LET R=10 AND X%*4+10 AND X%-1
+10 AND X%-1+10 AND X%*4+10 AND X%
+10+10 AND X%-1+10 AND X%*4+10
137 AND X%-1+10 AND X%*4+10 AND X%
+10 AND X%-1+10 AND X%*4+10 AND X%
+10 AND X%-1+10 AND X%*4+10 AND X%
+10
138 LET V=0: IF X%(4) OR X%(4) OR X%
+10 OR X%(4) OR X%(4) THEN
LET V=1
139 PRINT AT V,0: OVER 1: DRAWSE 11
: GOTO 1
139 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0
000 DATA 0,1,0,0,0,0,0,0,0,0,0,0,0
```

En las líneas 20 y 30 se dimensiona la matriz N(25) que contiene las notas asignadas a las teclas del piano, efectuándose su carga a partir de las líneas de DATA de las instrucciones 190 y 200

En las líneas 40 a 100, se dibujan las teclas «negras» del piano en las posiciones correspondientes. En 110 se imprimen las teclas que podemos pulsar y, sobre ellas, los correspondientes valores de las frecuencias de las notas. Debemos destacar el empleo de la función OVER, que permite la distinción de las letras impresas, en posiciones donde se solapan las teclas «blancas» y las «negras». Para terminar con este bloque, la línea 120

El empleo de OVER permite el resalte de las letras impresas, correspondientes a cada nota musical

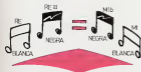
FA + OVER = FA



El sonido relacionado a la derecha de cualquier nota depende de la distancia de la misma.

tos, se brufca a la subrutina 160 encargada de encender y apagar la tecla pulsada en el lugar co rrespondiente del teclado

Esta es la parte más complicada del programa, procesamos la subrutina de 160 a 180. En la línea 160 se analiza el valor que se ha de dar a la variable X, lo cual representa el primer parámetro (horizontal) de la sentencia AT. Para ello, se recurre a la suma de las evaluaciones lógicas con relación de tipo AND, entre cada una de las posibilidades de teclas pulsadas, con el número de columna que les correspondería en la impresión



Las notas de las notas negras se cuentan apellidos: sostenido (S) y bemol (b)

Como sólo una de las evaluaciones puede ser correcta, la que lo sea dejará a X con el valor de la columna en la cual se debe realizar la impresión, produciendo el resto de las evaluaciones la suma de ceros, sin alterar el valor correcto de X. En la instrucción 170, se hace lo propio con el valor de la fila en la cual se ha de realizar la impresión. Sin embargo, esta vez se opta por un nuevo sistema, consistente en establecer de entrada el valor más frecuente, para ser modificado, caso de que no sea correcto, por la sentencia de comparación que lo sigue

Por último, en la línea 180, se efectúa la impresión por medio de AT, en la fila y columna especificadas por las variables X e Y. Esta impresión se realiza en OVER 1, de un cuadrado de color de la INK, por ello, la primera vez que se entra en la subrutina la letra se coloca en video inverso, y la segunda se pasa a video normal, apoyándose en las propiedades de sobreimpresión con OVER.

LOS GRANDES COMPOSITORES

A continuación, como muestra de las posibilidades de la sentencia BEEP en la composición de música, reproducimos el listado de un programa, que ejecuta algunos compases de la Marcha Turca de W. A. Mozart.



```

18 REM MARCHA TURCA - W.A.MOZART
20 FOR I=0 TO 259
30 READ A
48 BEEP ,84,A-2
50 FOR J=0 TO 31: NEXT J
60 NEXT I
70 DATA 11,9,8,9,12,12,12,12
80 DATA 14,12,11,12,16,16,16,16
90 DATA 17,16,15,16,20,21,20,21
100 DATA 20,21,20,21,24,24,24,24
110 DATA 21,21,24,24,20,20,21,21
120 DATA 19,19,21,21,20,20,21,21,19,19,
21,21,20,20,21,21,19,19,18,18,16,16
130 DATA 69,69,69
140 DATA 11,9,8,9,12,12,12,12
150 DATA 14,12,11,12,16,16,16,16
160 DATA 17,16,15,16,20,21,20,21
170 DATA 20,21,20,21,24,24,24,24
180 DATA 21,21,24,24,20,20,21,21
190 DATA 19,19,21,21,20,20,21,21,19,19,
21,21,20,20,21,21,19,19,18,18,16,16
200 DATA 69,69,69
210 DATA 16,16,17,17,19,19,19,21,19,
17,16,14,14,2,2
220 DATA 16,16,17,17,19,19,19,21,19,
17,16,14,14,2,2
230 DATA 12,12,14,14,16,16,16,16,17,16,
16,12,11,11,-1,-1
240 DATA 12,12,14,14,16,16,16,16,17,16,
16,12,11,11,-1,-1
250 DATA 11,9,8,9,12,12,12,12
260 DATA 14,12,11,12,16,16,16,16
270 DATA 17,16,15,16,20,21,20,21
280 DATA 20,21,20,21,24,24,24,24
290 DATA 21,21,20,20,24,24,23,21,21,
20,20,21,21
300 DATA 16,16,17,17,14,14,12,12,12,12
310 DATA 11,12,11,9,8,9,9,9,9,9

```

!

En los gráficos de las ondas sinusoidal y cuadrada, la ordenada expresa el «pico» y la abscisa el «tiempo». Imágenes «cristales» para de mayor altura y «velocidad» de menor

*

En el teclado de un piano existen un conjunto de teclas blancas correspondientes a las notas Do Re Mi, Fa Sol, La y Si (G, D, E, F, G, A, B) en notación musical interna, con las que se constituyen una «octava», y a las que corresponden ciertas tonas o frecuencias

*

Se denomina «amplitud» de onda a la máxima distancia entre una cresta o un valle y el que de abscisas, y «longitud» de onda a la distancia entre dos crestas o dos valles consecutivos

TURBO TRANSMISION



FRECUENTEMENTE, al interrogar a cualquier usuario del Spectrum sobre su experiencia manejando el cassette, contestará, casi con toda seguridad: «Es lento». La queja es, en cierto modo, justificada, aunque otros equipos similares llegan a agotar la paciencia durante las labores de carga y grabación, al utilizar velocidades de transmisión todavía menores.

Debe estar claro para todos, en este momento, que la velocidad de transmisión de los datos entre el Spectrum y el cassette no depende en absoluto de la rapidez de giro de la cinta. Es más, si por alguna razón el armazón de la cinta por el motor sufre fluctuaciones, puede ocasionar fallos de interpretación por parte de nuestro microordenador al variar las frecuencias correspondientes al caso y uno lógico.

Dicha velocidad de transmisión está controlada por las rutinas implementadas en la R. O. M. del Sistema. Al estar almacenadas en la memoria de sólo lectura, son fijas e inalterables por parte del usuario, pero como más adelante veremos, podemos actuar con habilidad sobre ellas, si previamente las hemos consultado en la memoria R. A. M., donde procederemos a modificarlas.

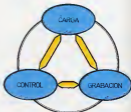


Procesamiento al presentar a cualquier usuario del Spectrum sobre el manejo de un cassette, contestará: «Es lento».

referencias a las direcciones y valores almacenados en ellas, en su formato decimal.

La entrada principal de esta zona del firmware se encuentra en la dirección 1541, donde comienza la subrutina SAVE-ETC (grabar-etc), punto común utilizado por los cuatro comandos del BASIC, SAVE, LOAD, MERGE y VERIFY. La primera parte está destinada a la creación de la cabecera de información en el espacio de trabajo

Las rutinas de manejo del cassette almacenadas en la ROM están divididas en tres grupos: grabación, carga y control, que ejecutan funciones similares entre sí.



ESCRITO EN LA R. O. M.

Las rutinas de manejo del cassette, situadas en la R. O. M. entre las direcciones decimales 1218 y 2547, están divididas en once subrutinas, que podemos agrupar en tres grandes grupos: grabación, carga y control de estas operaciones. Como ya estudiaremos más adelante, no son independientes entre sí, es decir, se efectúan constantes llamados de uno a los otros, y algunas pueden ser comunes a varios comandos.

Las rutinas que estamos tratando se encargan de gestionar cualquier operación que involucre el uso de instrucciones relacionadas con el almacenamiento de datos en cassette. A partir de ahora, y siempre que no se indique lo contrario, nos

!

La subrutina MCDINTR, ejecuta la orden de MERGE en tres etapas: copia del bloque de datos, modo de las líneas de los dos programas, y modo de los variables.

*

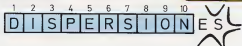
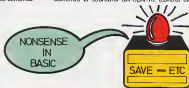
Entre las múltiples funciones que realiza la subrutina LD-BYTES, se encuentra la separación de los operaciones de carga y verificación, comparando los bytes uno a uno.

Primeramente, evalúa el tipo de comando, o construcción, reserva decimales localizaciones de memoria en el caso de **SAVE**, y treinta y cuatro para los otros tres.

Allí se almacenarán los datos concernientes al nombre de la información, longitud, etc (ver comentario del programa **INDEX**). La razón por la cual al efectuar **LOAD**, **MERGE** o **VERIFY**, se reservan otros decimales bytes, queda justificada por la necesidad de comprobar, en estas operaciones, si la cabecera que nosotros generamos tras uno de estos comandos coincide con la recibida desde la cinta. Caso de no ser así, se emite el correspondiente mensaje de error o permanente a la espera de recibir una nueva cabecera. Entre las funciones de la mencionada rutina de entrada, está comprobar si la longitud del nombre es mayor que diez caracteres, en cuyo caso emite el mensaje **F Invalid file name** (nombre no apto), si el comando era **SAVE**, o desprecia los caracteres en exceso, si se trata de cualquiera de los otros tres. Se comporta así mismo si la cadena nombre es nula, no aceptándolo en el caso de grabación, pero sí en la de lectura.

Por otra parte, la rutina **SAVE ETC** no admite la posibilidad de **MERGE**, cuando se trata de bloques de bytes, palabras o datos, comprueba si un programa ha sido almacenado con autoexclusión mediante **LINE**, evitando igualmente la opción de mezcla en este caso. Otra de sus misiones es imprimir en la pantalla el mensaje apropiado sobre el tipo de lectura: **Program**, **Bytes**, **Number array** o **Character array**, seguido del nombre, y finalmente, si todo fue bien, da paso a las subrutinas que gestionan cada comando.

SAVE ETC comprueba si la longitud del nombre introducido es mayor de 10 caracteres. En ese caso, si el comando es **SAVE** emite el mensaje **Nonsense in Basic** (sin sentido) o **Invalid file name**.



La ROM se ejecutará las rutinas de manejo del casete

SUBROUTINAS DE GRABACION

Una vez verificadas todas las funciones anteriores, si estamos efectuando una orden **SAVE**, la rutina anterior salta a la dirección 2 416, donde comienza la subrutina **SA CONTROL** (control de



Las rutinas implementadas en **ROM** controlan la velocidad, primer problema: actuar sobre este reubicándose en la memoria **RAM**.

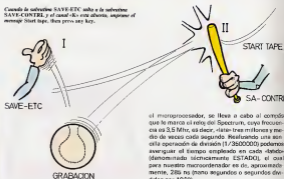
*

Las rutinas de control del casete residen en la **ROM** del ordenador.

*

Las rutinas de manejo de casete están divididas en tres grandes grupos: grabación, carga y control.

Cuando la subrutina SAVE-ETC sale a la subrutina SAVE-CONTROL y el canal -K- está abierto, imprime el mensaje Start tape, then press any key.



!

SAVE-ETC comprueba si la longitud del nombre introducido es mayor de 10 caracteres. En este caso, si el comando es SAVE entra un mensaje de error, si se trata de cualquiera de los otros tres se desprecian los caracteres en exceso.

*

La velocidad de grabación se obtiene como promedio, ya que el número de ceros y unos es prácticamente el mismo compensándose los tiempos de grabación.

*

La rutina LD-BLOCK trata como misas en seguir a la información que recibe pertenece a una cabecera o a un bloque de datos y fijar los colores rojo y cyan del borde de la pantalla.

grabación). Esta se asegura que el canal -K-, correspondiendo al teclado, está abierto a imprimir el mensaje Start tape, then press any key (ponga en marcha el casete y pulse cualquier tecla), que señala el comienzo de toda grabación en cinta, para posteriormente esperar la pulsación de una tecla.

Al hacerlo, realiza dos llamados sucesivos a la subrutina SA-BYTES (grabar bytes), ubicada a partir de la dirección 1218. En la primera de ellas envía los 17 bytes que componen la cabecera, a los cuales añade a su comienzo, a modo de bandera, un código 0 (cero) que indica el tipo de información que sigue (cabecera), y por dentro un byte de paridad. A continuación, produce el espacio muerto entre los tonos gutos, busca la longitud y comienzo del bloque de datos a ser grabado, y realiza la segunda llamada a SA-BYTES. En este caso, el primer byte enviado contendrá un 255, el cual indica el comienzo del bloque principal de información.

Esta última subrutina es la encargada, propiamente, de grabar el bloque de datos. Genera el tono guía previo a la emisión hacia el casete de la cabecera, graba esta, produce el segundo tono guía, y por último, graba el bloque de datos, descomponiendo cada uno de los bytes en bits individuales. Estos son enviados en serie, uno tras otro, hacia el casete. Además, es la encargada de fijar los colores en las bandas del borde de la pantalla, mientras se está ejecutando el proceso. Antes de continuar con el estudio de la importante rutina SA-BYTES, precisamos ampliar nuestro vocabulario informático con la palabra ESTADO. Como sabemos, cualquier operación que ejecuta

el microprocesador, se lleva a cabo al compás que le marca el reloj del Spectrum, cuyo frecuencia es 3,5 Mhz, es decir, «late» tres millones y medio de veces cada segundo. Realizando una sencilla operación de división (1/350000) podemos averiguar el tiempo empleado en cada «latido» (denominado técnicamente ESTADO), el cual para nuestro microcondensador es de, aproximadamente, 286 ns (nano segundos o segundos divididos por 1079).

Pues bien, cuando la subrutina SA-BYTES trata que enviar ceros y unos hacia el casete, lo hace siempre de la misma manera: mantiene la señal que sale por MIC en alto durante 855 estados, y en bajo durante otros 855 estados, si está enviando un cero, mientras que si el bit a grabar es un

Cualquier operación que ejecute el microprocesador, se divide y cede al compás que le marca el reloj del Spectrum que late a 3,5 Mhz.



uno, sigue el mismo proceso, pero a los 855 segundos en alto-bajo, les añade otros 855 segundos. Por ello, decimos que la frecuencia del caso es doble a la del uno Kílogos.

Este sistema de grabación da pie a un hecho curioso: cualquier zona de memoria, por ejemplo una pantalla, llena de ceros tarda la mitad de tiempo en ser grabada o cargada que una llena de unos [255]. Así pues, la velocidad de grabación es un promedio, dado que estadísticamente, el número de ceros y unos en cualquier grabación es prácticamente igual y, por tanto, sus tiempos de grabación se compensan.

SUBROUTINAS DE CARGA

Una vez comparadas las cabeceras, la indicada por nosotros tras un comando **LOAD**, **VERIFY** o **MERGE** y la que proviene de la cinta, en caso de coincidencia, la rutina **SAVE-ETC** pasa a considerar, independientemente, cada una de los comandos.

Si la orden efectuada fue **LOAD**, salta a la subrutina **LD-CONTRL** (control de carga), ubicada a partir de la dirección 2.056. Es la encargada de supervisar la carga de un programa **BASIC** y sus variables, así como variables susoditas (matrices). Comienza considerando si se trata de un programa **BASIC** o de una matriz. En el primer caso, emite el mensaje **4 Out of memory**, si no dispone de suficiente espacio en la memoria para almacenarlo. Si lo hay, lo reserva, de manera que tengan cabida tanto el programa como sus variables. Finalmente, salta a la subrutina **LD-BLOCK** (cargar bloque) ubicada entre las direcciones 2.050 y 2.055.

Esta subrutina es común en todos los procesos que impliquen la carga de cualquier bloque de información. Su misión consiste en realizar una llamada a la subrutina **LD-BYTES** (carga de bytes)

La subrutina **LD-BYTES** agota las funciones de carga y verificación.



Si el nombre de la cinta sigue las convenciones, puede aparecer fallas de interpretación en nuestro ordenador.

que ocupa las direcciones de la **R O M** comprendidas entre la 1.366 y la 1.506. Esta rutina es llamada dos veces: una para cargar la cabecera de información y otra para los bloques de datos.

Su misión consiste, como primer paso, en averiguar si la información que recibe pertenece a una cabecera o a un bloque de datos. Además, se hace distinción entre los comandos **LOAD** y **VERIFY**. A continuación, toma un byte del port 254 (**FAR**), y lo interpreta bit a bit, lo cual constituye la lectura de datos propiamente dicha. Es también la encargada de fijar los colores rojo y cyan que observamos en el borde de la pantalla, cuando la subrutina, tras ejecutar cualquier comando de carga, está a la espera de recibir la información.

Está preparada para recibir la información contenida en la cabecera primero, y después, tras el segundo tono guía, la del bloque principal de datos. Para ello se realizan varias llamadas a otras subrutinas, denominadas **LD-EDGE-1** y **LD-EDGE-2**, comprendidas entre las direcciones 1.507 y 1.540. La traducción podría ser «cargar bordes», quizás haciendo referencia a que conforme se va recibiendo la información, las bandas laterales de la pantalla van variando según se trate de unos, de ceros o del tono guía para el sincronismo de cabecera.

Estas últimas rutinas son las más importantes en todo proceso en el cual intervenga la carga o verificación de un bloque de datos. Son las encargadas de medir la duración de un pulso completo, así como el tiempo empleado antes de la siguiente transición entre los estados en que el pulso se encuentra alto o bajo. En las páginas an-

!

SA-BYTES es la subrutina encargada de grabar el bloque de datos, generar los tonos guía, descomponer los bytes en bits, y fijar los colores en las franjas del borde de la pantalla.

*

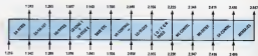
El punto de entrada común utilizado por **SAVE**, **LOAD** y **VERIFY** es la subrutina **SAVE-ETC**.

*

La rutina **SA-CONTRL** se asegura de que el canal del teclado está abierto emitiendo el mensaje **Start tape**, then press any key.

*

SAVE-ETC no admite el **MERGE** de bloques de bytes, postales o datos.



Mapa de valores de ROM para el acceso de datos

tericas, bajo el epígrafe «transmisión de datos», se dan las explicaciones suficientes sobre el aspecto de los pulsos correspondientes al cero, uno, y cabecera de transmisión.

La subrutina LD BYTES, completa su trabajo separando las funciones de carga y verificación. Durante esta última operación, compara los bytes que recibe de la cinta, uno a uno, con los almacenados en la memoria, emitiendo el mensaje **R Tape loading error** (error de lectura) cuando no coinciden ambos. Cabe aclarar, que los bytes recibidos no se almacenan en la memoria **R. A. M.**, sólo que se comparan directamente.

La supervisión de este proceso recae en la subrutina **VR-CONTROL** (control de verificación), ubicada entre las direcciones 1 995 y 2 049. Es utilizada, además, como rutina de control en la carga de programas que han sido grabados con las especificaciones **SCREEN*** o **CODE**. Es decir, por ejemplo, para recuperar una pantalla, el proceso sería el mismo descrito anteriormente para un programa **BASIC**, sólo que las rutinas de control serían diferentes.

SUBROUTINAS DE MEZCLA

Cuando efectuamos una orden **MERGE** (mezcla) la rutina **SAVE ETC**, tras realizar todos los pasos señalados anteriormente, entre ellos, comprobar

si se trata de un programa **BASIC** (sólo a **ME-CONTROL** (control de mezcla), situada a partir de la dirección de memoria 2 230).

Esta subrutina se ejecuta en tres etapas independientes:

1) Carga el bloque de datos en el espacio de trabajo.

2) Mezcla las líneas del nuevo programa, con el antiguo presente en la memoria.

3) Mezcla las variables.

La primera parte, reserva el espacio de memoria suficiente para albergar el programa, y realiza una llamada a la subrutina **LD-BLOCK**, encargada de cargar el nuevo bloque de datos en este espacio.

Durante el proceso de unión, compara uno a uno, los números de línea entre los programas nuevo y antiguo. Si coinciden, la dirección de comentario, tanto de la antigua como de la nueva será la misma. A continuación, averigua la dirección de memoria donde introducir la nueva. En ese momento realiza una llamada a la subrutina **ME-ENTER**, ubicada entre las direcciones 2 348 y 2 415, la cual se encarga de fusionar tanto las líneas de programa, como las variables. Por ello, cuando ejecutamos una orden **MERGE**, si existen números de líneas idénticas en los dos programas, serán los del nuevo los que prevalezcan.

Cuando ha terminado de mezclar todas las líneas, comienza un proceso similar con las variables de ambos programas. Compara los nombres byte a byte, y de ser iguales, el contenido de la nueva quedará almacenado, perdiéndose el de la antigua. Una llamada a **ME-ENTER**, cada vez que sustituye o añade una variable, completa el proceso. Finalmente, hay que indicar que la subrutina **SA/LD-RET** (1 343 a 1 365), retorno de cargar/salvar, es común a ambos procesos, devolviendo al borde su color original y leyendo por última vez a la tecla **BREAK** fue pulsada. Tras ella,

ME-ENTER se encarga de fusionar tanto las líneas de programa como las variables.



SA-BYTES es la encargada de fijar los colores en los bandos del fondo de la pantalla



se devuelve el control al BASIC al terminar el uso de las rutinas de manejo del casete. Entre los direcciones 2 465 y 2 547, están almacenados los mensajes que aparecen durante los procesos relativos al almacenamiento de información en la grabadora, tales como *Start tape, then press any key, program, etc*

REUBICANDO

El programa «Reubicador Turbo» que aparece en las próximas páginas, transfiere la zona de la R. O. M. que contiene todas las rutinas comentadas anteriormente, menos la tabla de mensajes, a la parte de la R. A. M. escogida por nosotros. Una vez aquí, se efectúan los cambios destinados a modificar la velocidad de transmisión de los datos entre Spectrum y casete. En primer lugar, hemos realizado una nueva lista de los mensajes relacionados con las opera-

ciones de casete. Los hemos traducido al castellano y podemos utilizar cualesquiera otros, siempre y cuando respetemos su formato. Pero deberemos considerar la longitud de los mismos, pues el tamaño del código máquina a grabar en la última línea del programa TURBO-GEN (reubicador turbo), dependerá del número de caracteres ocupado por dichos mensajes.

Las líneas **DATA** que contienen valores numéricos son las que encierran el verdadero secreto del método de transmisión de datos a velocidad variable. Cada una está formada por parejas de valores: el primero es una dirección y el segundo el nuevo valor almacenado en ella. La primera línea corresponde a las constantes de tiempo que influyen durante el proceso de grabación, y las siguientes a las relacionadas con las operaciones de carga.

A continuación, se procede a reubicar todas las instrucciones en código máquina de las rutinas de manejo del casete que comportan saltos absolutos o llamados a otras subrutinas. No queda otro remedio que hacerlo así, pues al estar nuestra rutina ubicada en la R. A. M., los saltos y llamadas deben ser entre direcciones de ésta para asegurar su perfecto funcionamiento.

Finalmente, los **POKE** K+1 935 y K+1 936, son los encargados de fijar el nuevo origen para los mensajes que aparecen en cualquier operación de manejo del casete. Si no actuamos sobre estas direcciones, será vano nuestro esfuerzo al tratar de conseguir, por ejemplo, el mensaje «Por en marcha la grabadora» y seguirá apareciendo el ya conocido, *Start tape, then press any key*. Pongamos en práctica nuestros conocimientos en la sección de PROGRAMA.

SA-BYTES

!!

La subrutina LD-CONTROL es la encargada de supervisar la carga de un programa BASIC, sus variables y acciones.

*

La frecuencia de grabación del D lógico es doble que la del 1.

*

La subrutina encargada de fijar los límites de programa y las variables de MENTER.



Para ser empleadas en ROM controlan la velocidad del casete, pero podemos actuar sobre ésta reubicándolas en la memoria RAM.

TURBO



A llegado la hora de que pongamos en práctica todos nuestros conocimientos sobre transmisión a alta velocidad (turbo-transmisión), y para ello utilizaremos los programas cuyas instrucciones de manejo detallamos a continuación.

Nuestro sistema de turbo-transmisión se fundamenta, como hemos visto en las páginas anteriores, en la realización de la zona de R. O. M. que contiene las rutinas de manejo del casete, para posteriormente realizar sobre ellas una modificación que aumenta la velocidad de transmisión, en nuestro caso, nada más y nada menos que al doble de la normal: 3.000 baudios.

Disponemos de dos programas bien diferenciados: el primero de ellos, denominado **TURBO-GEN**, realiza la función de reubicar las rutinas afectadas a una zona de la R. A. M., generando un código máquina que será posteriormente utilizado por el segundo programa (**TURBO-RUN**), cuya misión es proporcionar un apoyo BASIC para la cómoda utilización del sistema de turbo-transmisión.

TURBO-GEN

El manejo de **TURBO-GEN** es bien sencillo, casi podríamos decir que automático. El único dato que nos reclama, al comienzo de su ejecución, es el lugar donde queremos reubicar las rutinas de manejo del casete. Esto supone una considerable ventaja, dado que sitúa en el lugar de la R. A. M. que deseamos el código máquina que utilizará **TURBO-RUN**, y por tanto, proporciona una total flexibilidad en cuanto a los programas que podemos tratar por el sistema, emplazando el código máquina en el lugar de la memoria en que menos nos entorpezca.

Sólo hemos de tener en cuenta que no puede ser reubicado al buffer de la impresora, puesto que, como veremos más adelante, esta zona es empleada por el propio **TURBO-RUN** como zona de trabajo.

Una vez que hemos indicado la dirección de ubi-

cación, y tras un corto intervalo de tiempo, irá recibiendo el mensaje "PREPARE LA CINTA PARA GRABAR". Esto quiere decir que el código máquina ha sido situado en el lugar de la memoria que deseamos, y protegido convenientemente mediante **CLEAR**. Esta opción de grabación del código máquina generado, nos evitará sucesivos veces recurrir al programa **TURBO-GEN**.

Pese a todo, si por cualquier motivo no nos interesa grabarlo, podemos interrumpir el programa mediante **BREAK**, y borrar el soporte BASIC con **NEW**, en la certeza de que el código máquina se encuentra correctamente reubicado y protegido por el **RAMTOP**.

TURBO-RUN

El sistema empleado por **TURBO-RUN** es de una gran complejidad: se sitúa en una zona de trabajo (el buffer de la impresora) la instrucción BASIC que se desea ejecutar, y a continuación se efectúa una sencilla llamada al punto de reubicación de **SAVE-ETC**, con lo cual el resto del trabajo corre a cargo del firmware reubicado en la R. A. M.

Sabiendo esto, y con unos pequeños conocimientos de la estructura de las instrucciones BASIC,

MENÚ TURBO-GEN

GRABAR	(g)
VERIFICAR	(v)
CARGAR	(c)
MEZCLA	(m)

Este menú permite seleccionar entre las opciones: grabar, verificar, cargar y mezclar.

!

La turbo-transmisión se fundamenta en la reubicación de la ROM para realizar una modificación que aumenta la velocidad de transmisión en el doble.

*

Disponemos de dos programas para esto: **TURBO-GEN** (código máquina) y **TURBO-RUN** (apoyo BASIC).

no nos será nada complicado utilizar el código máquina de **TURBO-GEN** sin necesidad de tener que recurrir a **TURBO-RUN**.

Para aquellos a los que nos resultaría complicado generar la instrucción pertinente en el área de trabajo, el programa **TURBO-RUN** resalta el trabajo en nuestro lugar, asegurándose además de la correcta sintaxis del comando.

El programa **TURBO-RUN** debe ser ejecutado a partir de la línea 9987, es la cual le indicaremos el lugar donde se encuentra ubicado, o se va a ubicar, el código máquina. Ato seguido se aguar-

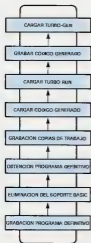
da la entrada del código máquina desde el casete. Si este se encuentra en memoria, debido a la reciente ejecución de **TURBO-GEN**, el programa puede ser ya interrumpido y se encuentra listo para el funcionamiento.

Llegados a este punto, podemos comenzar a programar nuestro propio BASIC, o bien efectuar un **MERGE** del programa que deseamos grabar a alta velocidad, teniendo en cuenta que el soporte de **TURBO-RUN** ocupa todas las líneas desde la 9900 a la 9993, ambas inclusive, por lo cual dichos números de instrucción no pueden ser empleados para nuestro propio uso.

La finalidad de nuestro sistema de turbo-transmisión es aumentar la velocidad con que utilizamos el casete para las copias de trabajo, de forma que una vez que hemos llegado a una versión definitiva de nuestro programa, podemos eliminar el soporte de **TURBO-RUN**, y grabar a velocidad normal, y más segura, la copia definitiva del programa.

Así pues, cuando grabamos programas en turbo, el soporte BASIC de **TURBO-RUN** se graba con ellos. ¿Cómo hemos de hacer para eliminarlo en la última copia? Dado que esta será efectuada a velocidad normal, no precisamos ninguna de las líneas de **TURBO-RUN**, si bien es cierto que eliminar, una por una, las 94 líneas que lo componen es muy pesado.

La solución se encuentra en la línea 9992, ésta contiene un breve código máquina que destruye todo vestigio de la presencia de **TURBO-RUN**. Sólo hemos de editar esta línea y suprimirle el número de instrucción mediante **DELETE**, para que sea ejecutada en forma de comando directo.



Organograma de utilización de los programas **TURBO-GEN** y **TURBO-RUN**.



En este menú podemos seleccionar que clase de datos deseamos almacenar o recuperar del casete.



Para obtener una copia de trabajo exacta antes a la línea 9900, seleccionando la opción adecuada.

*

Si no nos interesa grabar el código máquina, después de situado en la memoria, podemos interrumpir el programa para borrar el soporte BASIC con **NEW**.

Una vez que **TURBO-RUN** ha sido inicializado, es decir, se ha efectuado la ejecución a partir de 9881 y el código máquina se encuentra en la memoria, ya no necesitamos recurrir más veces a este punto hasta que no berremos la memoria del ordenador.

Cada vez que deseamos reanudar una copia de trabajo deberemos acceder a la línea 9800, donde se sucederán una serie de menús en los cuales podremos escoger de manera muy sencilla el comando que deseamos ejecutar.



TURBO-GEN selecciona el lugar donde recibir los rutinas de manejo del sistema



Al efectuar se **MENÚ** del programa que deseamos grabar a alta velocidad, no debemos olvidar las líneas del **9800** al **9880**



No podemos trabajar el buffer de memoria por nosotros **TURBO-RUN** como zona de trabajo



Si tenemos una versión definitiva de un programa y deseamos almacenar el soporte **BASIC** de **TURBO-RUN**, basta editar la línea **9882**, suprimiendo el número de instrucción para ejecutarla en comando directo

```

9800 REM DESCRIPCION TURBO - G DE LA OMSA S F. LINDA PA
RTIME:
9801 INPUT "SELECCION DE SELECCION " ; D
9802 RANDOMIZE D
9803 CLEAR D=1: LET B=HEX 25A0F-25A0FEEK 23A1
9804 FOR I=102376 TO 23087
9805 READ A: POKE 2, A
9806 NEXT I
9807 DATA 35,151,4,1,7,RED: 23A70,POK 20A71,1,230,1,2
37,7A,201
9808 RANDOMIZE USR 23214
9809 LET C=D-128: LET A=23A64K
9810 FOR I=0 TO 2
9811 READ B
9812 FOR J=1 TO LEN B
9813 POKE A+J, B
9814 NEXT J
9815 NEXT I
9816 DATA "Van en marcha la grabadora"
9817 DATA "TURBO-GEN 13"
9818 DATA "Materia seleccionada"
9819 DATA "Materia seleccionada"
9820 DATA "Materia seleccionada"
9821 DATA "Materia seleccionada"
9822 FOR I=0 TO 20
9823 READ A: B
9824 POKE A+I, B
9825 NEXT I
9826 DATA 2230,2,1212,25,1232,12,1271,14,1269,15
9827 DATA 2272,29,1282,30,1221,30,1224,70,1232,29
9828 DATA 1231,15,1282,2,1212,13
9829 DATA 1444,204,1479,210,1487,230,1490,214,1502,12
9830 DATA 1503,10
9831 DATA 1241,0
9832 FOR I=0 TO 20
9833 READ A
9834 LET B=HEX 475A6FEK 14410
9835 RANDOMIZE B: LET B=HEX 23A70: LET B=HEX 23
471
9836 RANDOMIZE B: LET B=HEX 23A70: LET B=HEX 23
471
9837 POKE B+25A64, B: POKE B+25A64+1, B: REPT -40
1
9838 NEXT I
9839 DATA 1221,1254,1276,1270,1209,1239,1270,1261,149
4,1481
9840 DATA 1434,1436,1483,1484,1506,1474,1487,1483,148
0
9841 DATA 1193,7851,1141,2229,7051,2294,2304,2420,214
3,2483
9842 POKE B+6703, B: POKE B+6706, B
9843 PRINT "MATERIA LA OMSA PAUL GREGG"
9844 GOTO 2
9845 DATA "C/M TURBO" "C/M 0,330"

```

```

9846 PRINT AT 0,0, "IMAGEN 1, 2", "IMAGEN 3"
"IMAGEN 4" "IMAGEN 5" "IMAGEN 6" "IMAGEN 7"
"IMAGEN 8" "IMAGEN 9" "IMAGEN 10" "IMAGEN 11"
"IMAGEN 12" "IMAGEN 13" "IMAGEN 14" "IMAGEN 15"
"IMAGEN 16" "IMAGEN 17" "IMAGEN 18" "IMAGEN 19" "IMAG
EN 20"
9847 FOR I=23070 TO 23079
9848 LET B=HEX 23070: POKE B, B
9849 IF B=HEX 23070 THEN GO TO 9850
9850 INPUT "MATERIA OMSA " ; LINE TA: "T" THEN
LET TA="T"
9851 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9852 IF NOT MATRIZ=HEX 00 GO TO 9853
9853 INPUT "MATERIA OMSA " ; LINE TB: "T" THEN
LET TB="T"
9854 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9855 IF NOT MATRIZ=HEX 00 GO TO 9856
9856 INPUT "MATERIA OMSA " ; LINE TC: "T" THEN
LET TC="T"
9857 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9858 IF NOT MATRIZ=HEX 00 GO TO 9859
9859 INPUT "MATERIA OMSA " ; LINE TD: "T" THEN
LET TD="T"
9860 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9861 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9862 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9863 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9864 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9865 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9866 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9867 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9868 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9869 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9870 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9871 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9872 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9873 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9874 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9875 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9876 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9877 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9878 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9879 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9880 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9881 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9882 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9883 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9884 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9885 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9886 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9887 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9888 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9889 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9890 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9891 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9892 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9893 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9894 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9895 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9896 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9897 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9898 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9899 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9900 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9901 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9902 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9903 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9904 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9905 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9906 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9907 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9908 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9909 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9910 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9911 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9912 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9913 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9914 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9915 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9916 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9917 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9918 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9919 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9920 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9921 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9922 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9923 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9924 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9925 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9926 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9927 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9928 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9929 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9930 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9931 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9932 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9933 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9934 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9935 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9936 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9937 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9938 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9939 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9940 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9941 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9942 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9943 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9944 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9945 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9946 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9947 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9948 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9949 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9950 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9951 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9952 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9953 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9954 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9955 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9956 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9957 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9958 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9959 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9960 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9961 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9962 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9963 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9964 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9965 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9966 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9967 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9968 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9969 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9970 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9971 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9972 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9973 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9974 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9975 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9976 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9977 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9978 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9979 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9980 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9981 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9982 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9983 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9984 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9985 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9986 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9987 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9988 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9989 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9990 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9991 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9992 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9993 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9994 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9995 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9996 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9997 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9998 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852
9999 IF MATRIZ=HEX 00 THEN IF MATRIZ=HEX 00 MATR
IZ=HEX 00: GO TO 9852

```